Advanced Data Analytics – Assignment 1

Since this is my first Data Science course and I had hardly had any contact with the topic before at my Home University in Germany, it took me a long time to read up on the topic. Another problem was that the dropbox link for the test and training data was no longer accessible.

After analyzing the code and doing a lot of research on what the individual lines mean, I started to get a better overview of the test and training data. Therefore I executed *xy_train.head()* to look at the first 5 lines of the training sample and then I executed *xy_train.describe()* to get a better overview. This gave me the knowledge about which columns are available and first ideas which could be relevant. The histogram also showed that most of the apartments were in the cheaper segment, some in the medium and few in the expensive segment.

Now that I knew what the data looked like, I continued to try new models to increase the accuracy. Because the house price was divided into 3 categories instead of an amount, the linear regression and logistic regression models make no sense. Besides the given XGB classifier, I tried random forest and MLP classifier. All these models work well with categorical and numerical features.

MLP optimizes the log-loss function using LBFGS or stochastic gradient descent. Here I came to a result of: 73.29% (after adding features).

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Here I got an accuracy of: 73.81% (after adding features).

Since both new models showed hardly any improvement, I wanted to try to use ensemble methods to combine the models to get a better accuracy. Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to improve predictions. For this purpose the models are packed into a voting classifier and the best result is chosen by voting. In ensemble algorithms, bagging methods form a class of algorithms which build several instances of a black-box estimator on random subsets of the original training set and then aggregate their individual predictions to form a final prediction. This led me to an accuracy of over 90%.

To further improve this, I decided to add more numerical and categorical features to give more information to the models. After trying out some features and leaving some out, I came up with a final accuracy of: **93.18%**

This is an improvement of over 23% compared to the beginning. My model will probably score worse on the public and private leaderboard, because I guess my model overfits the training data a bit. To improve this, you would have to have more training data or you would have to add some of the test data to the training data, because it is currently a 50/50 distribution, but a 70/30 distribution would probably be better and more common.

Questions 1: Why should we limit the number of trials per day?

We should limit the number of attempts, so that you don't, for example, try out all features endlessly until you get the best result, but choose your features wisely instead of trying them out randomly. This also applies to other code changes, you should stick to the Data Science lifecycle and make conscious decisions instead of just trying everything without knowing until you get the best fitting model.

Question 2: Why it is the private leaderboard designed like this?

I think it is designed that way, because the test data for the public leaderboard should act as a validation set and then the test set for the private leaderboard should represent real data.

Question 3:

One model I used was random forest. The flexibility here is controlled by several hyperparameters. It is possible to play with the number of trees to make it more flexible. You can also play with the depth to limit it and not make it too flexible. Otherwise it would become too specific and give too much attention to single parameters.